

# 詳細セットアップ手順

## ステップ1: プロジェクトファイルの準備

### 1-1. ディレクトリ構造を作成

```
bash
mkdir ec-training-site
cd ec-training-site
mkdir -p src/pages src/services public
```

### 1-2. 各ファイルを配置

以下のファイルを作成し、提供されたコードを貼り付けてください:

#### ルートディレクトリ:

- package.json
- README.md

#### public/ ディレクトリ:

- index.html

#### src/ ディレクトリ:

- index.js
- index.css
- App.js
- App.css

#### src/services/ ディレクトリ:

- api.js

#### src/pages/ ディレクトリ:

- ProductListPage.js
- ProductListPage.css
- CartPage.js
- CartPage.css

- CheckoutPage.js
- CheckoutPage.css
- OrderHistoryPage.js
- OrderHistoryPage.css

## ステップ2: Google Apps Script のセットアップ

### 2-1. スプレッドシートの作成

1. Google Drive にアクセス
2. 「新規」 → 「Google スプレッドシート」 → 「空白のスプレッドシート」
3. タイトルを「ECサイト研修DB」に変更

### 2-2. シートの作成

スプレッドシート下部の「+」 ボタンで6つのシートを作成:

1. products
2. members
3. cart\_items
4. orders
5. order\_items
6. coupons

### 2-3. スプレッドシートIDの取得

URLから以下の部分をコピー:

<https://docs.google.com/spreadsheets/d/【ここがID】/edit>

例: 1ABCDfGHljkLMNOpqrSTUVwxyZ123456789

### 2-4. Apps Script エディタを開く

1. スプレッドシートで「拡張機能」メニュー
2. 「Apps Script」 をクリック
3. 新しいタブでエディタが開きます

## 2-5. GASコードを貼り付け

1. デフォルトの `function myFunction() {}` を全て削除
2. 提供された「Google Apps Script - ECサイトAPI」コードを全てコピー
3. エディタに貼り付け
4. 2行目の `SPREADSHEET_ID` を実際のIDに変更:

```
javascript
```

```
...const SPREADSHEET_ID = '1ABCDDefGHIjklMNOpqrsTUVwxyZ123456789';
```

5. `Ctrl+S` (Mac: `Cmd+S`) で保存

## 2-6. 初期データの投入

1. エディタ上部の関数選択ドロップダウンをクリック
2. `initializeData` を選択
3. 「実行」ボタン(▶)をクリック
4. 初回実行時の権限承認:
  - 「権限を確認」をクリック
  - Googleアカウントを選択
  - 「このアプリは確認されていません」と表示されたら「詳細」をクリック
  - 「(プロジェクト名)に移動」をクリック
  - 「許可」をクリック
5. 実行ログに「Execution completed」と表示されればOK
6. スプレッドシートを確認し、各シートにデータが入っていることを確認

## 2-7. ウェブアプリとしてデプロイ

1. エディタ右上の「デプロイ」ボタンをクリック
2. 「新しいデプロイ」を選択
3. 「種類の選択」(歯車アイコン)をクリック
4. 「ウェブアプリ」を選択
5. 設定を入力:
  - 説明: `ECサイトAPI v1` (任意)
  - 次のユーザーとして実行: `自分(あなたのメールアドレス)`

- アクセスできるユーザー: **全員** (重要!)

6. 「デプロイ」をクリック

7. 「ウェブアプリのURL」をコピーして保存

例: `https://script.google.com/macros/s/AKfycbz.../exec`

## 2-8. 動作確認

ブラウザで以下のURLにアクセス:

`https://script.google.com/macros/s/あなたのID/exec?path=products&method=GET`

以下のようなJSONが返ってくればOK:

```
json
{
  "success": true,
  "data": [
    {
      "id": 1,
      "name": "JavaScript入門",
      "price": 2800,
      ...
    }
  ]
}
```

## ステップ3: Reactアプリのセットアップ

### 3-1. Node.jsのインストール確認

```
bash
node --version
npm --version
```

インストールされていない場合は [Node.js公式サイト](#) からダウンロード

### 3-2. 依存関係のインストール

```
bash
```

```
cd ec-training-site
npm install
```

### 3-3. API URLの設定

`src/services/api.js` を開き、3行目を編集:

```
javascript

// 変更前
const GAS_API_URL = 'YOUR_GAS_DEPLOYMENT_URL_HERE';

// 変更後
const GAS_API_URL = 'https://script.google.com/macros/s/あなたのID/exec';
```

### 3-4. ローカルで起動

```
bash

npm start
```

ブラウザが自動的に開き、`http://localhost:3000` でアプリが表示されます。

### 3-5. 動作確認

1. 商品一覧が表示されるか確認
2. 商品をカートに追加
3. カートページで数量変更・削除
4. チェックアウトで注文
5. 注文履歴で確認

## ステップ4: GitHub Pagesへのデプロイ

### 4-1. GitHubリポジトリの作成

1. GitHub にログイン
2. 右上の「+」 → 「New repository」
3. Repository name: `ec-training-site`
4. Public を選択
5. 「Create repository」をクリック

## 4-2. ローカルリポジトリの初期化

```
bash
cd ec-training-site
git init
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/YOUR_USERNAME/ec-training-site.git
git push -u origin main
```

## 4-3. package.json の編集

`package.json` の2行目の `homepage` を編集:

```
json
{
  "name": "ec-training-site",
  "homepage": "https://YOUR_USERNAME.github.io/ec-training-site",
  ...
}
```

`YOUR_USERNAME` を実際のGitHubユーザー名に置き換え

## 4-4. gh-pages のインストール

```
bash
npm install --save-dev gh-pages
```

## 4-5. デプロイ

```
bash
npm run deploy
```

数分待つと、以下のURLでアクセス可能:

```
https://YOUR_USERNAME.github.io/ec-training-site/
```

## 4-6. GitHub Pages の設定確認

1. GitHubリポジトリページを開く
2. 「Settings」 → 「Pages」

3. Source が `gh-pages` ブランチになっていることを確認

## トラブルシューティング

### Q1: GASで「ReferenceError: SpreadsheetApp is not defined」

A: Apps Scriptエディタで実行してください(通常のJavaScriptコンソールでは実行できません)

### Q2: 商品が表示されない(モックデータが表示される)

A: `src/services/api.js` の `GAS_API_URL` を実際のURLに変更したか確認

### Q3: CORSエラーが出る

A: GASのデプロイ設定で「アクセスできるユーザー」が「全員」になっているか確認

### Q4: npm start でエラー

A: 以下を試してください:

```
bash
rm -rf node_modules package-lock.json
npm install
npm start
```

### Q5: GitHub Pagesで404エラー

A:

1. package.json の homepage が正しいか確認
2. `npm run deploy` を再実行
3. 数分待ってから再度アクセス

## 次のステップ

システムが正常に動作したら:

1. **テストケースの作成:** 提供されたテンプレートを使用
2. **バグの発見:** 意図的にバグを仕込んで発見演習
3. **機能追加:** レビュー機能、お気に入りなど
4. **テスト自動化:** Playwright や Cypress でE2Eテスト

## サポート

質問がある場合は、GitHubのIssueで質問してください!